Abstract

The applicability of the *component-based design* approach to the design of internal combustion engines is demonstrated by developing a simplified model of such an engine under automatic speed control, in the Ptolemy environment. A key feature of such a model is its modular construction, which allows sub-models of different kinds to be connected together to form a realistic representation of this complex device. It also allows the individual modules to be "changed out" with others that are more or less refined, depending on the requirements of the task at hand. Finally, such a model provides a framework in which the different engineering disciplines involved can contribute in an effective and

Proposed Approach/Method

This project focuses on the development of a hybrid model of an engine that can provide information on the engine's state at any instant. In order to accomplish this we'll take advantage of the *hybrid systems* and *concurrent computing* theory that has come of age in the last 10-15 years, as implemented on Ptolemy. This provides a much more convenient environment in which to construct such models than the traditional computing environment. A rich and extensive literature has already been developed on the engine-modeling problem, and a bibliography of some of the more pertinent articles is given at the end of this report. Here we borrow and modify the results of several of these articles ([4], [7], [9], [14], [15], and [23]), to develop and implement our own model in the Ptolemy environment. Since the goal here is to demonstrate the method, the analytical sub-models used are kept relatively simple. They are still complex enough, though, to give realistic results and also to demonstrate the model's and Ptolemy's robustness.

The Four Stroke Cycle and Piston Position

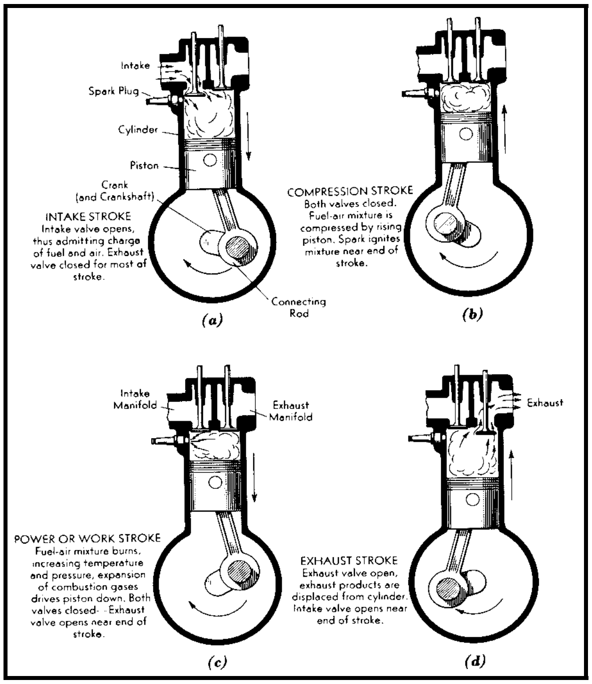
Figure 1 describes the thermodynamic cycle that a four-stroke spark ignition engine undergoes. The cycle consists of four "strokes" of the piston (intake, compression, combustion/expansion, and exhaust), each taking a half revolution of the crankshaft, and each beginning and ending at either the piston's *top dead center (TDC)* or *Bottom Dead Center (BDC)* position (its highest or lowest position). It thus takes two revolutions of the crankshaft (720°) to complete one cycle. If the vehicle is moving forward (which we assume throughout), its crankshaft angular velocity, ω_{cs}, will be positive and its angular position ϕ_{cs} will simply be the integral of ω_{cs}, a monotonically increasing function in time. ϕ_{cs} modulo 720 will then be a "saw tooth" waveform that increases from zero to 720° (an example, to be discussed later, is shown in Figure 22b.)

The cylinder delivers a positive torque in only one of these strokes—the expansion stroke. In all of the others it requires torque. The torque output of a cylinder is thus quite variable and fluctuates between positive and negative values. A real engine consists of more than one cylinder, and these are *offset* on the crankshaft with respect to one another to smooth out the engine's overall output. Figure 2 indicates how this *offset angle* ϕ_{co} is defined—positive when the piston is offset counterclockwise from $\phi_{cs}=0$ and negative when it is offset clockwise from it. The *piston angle* ϕ_p is then defined as

$$\phi_p \equiv \phi_{cs} - \phi_{co} \qquad -180° \le \phi_{co} \le +180°, \tag{1}$$

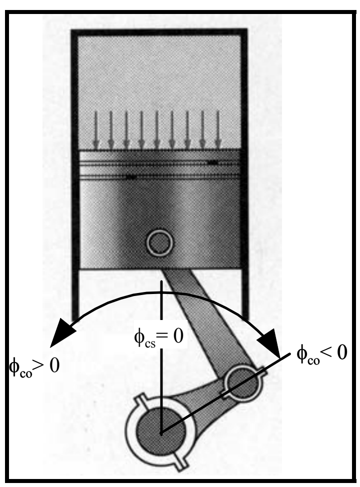
which is a convenient way to specify the piston's position. In the simplest case, when $\phi_{co}=0$, $0 \le \phi_p = \phi_{cs} \le 720$ and piston position is specified according to the convention

$$\begin{aligned} 0° \le \phi_p \le 180° \quad &\Rightarrow \quad \text{Intake,} \\ 180° < \phi_p \le 360° \quad &\Rightarrow \quad \text{Compression,} \\ 360° < \phi_p \le 540° \quad &\Rightarrow \quad \text{Expansion,} \\ 540° < \phi_p \le 720° \quad &\Rightarrow \quad \text{Exhaust.} \end{aligned} \tag{2}$$



The Four-Stroke Cycle

Figure 1[1]



The Piston Offset Angle ϕ_{co}

Figure 2[2]

---

[1] Figure taken from "Internal Combustion Engines", by E. F. Obert, International Textbooks, 1968, p. 3.

[2] Figure taken from "Thermodynamics: An Engineering Approach", by Y. A. Çengel & M. A. Boles, McGraw-Hill, 1994, p. 15.

The more general case is a little more complicated and is presented in Table 1.

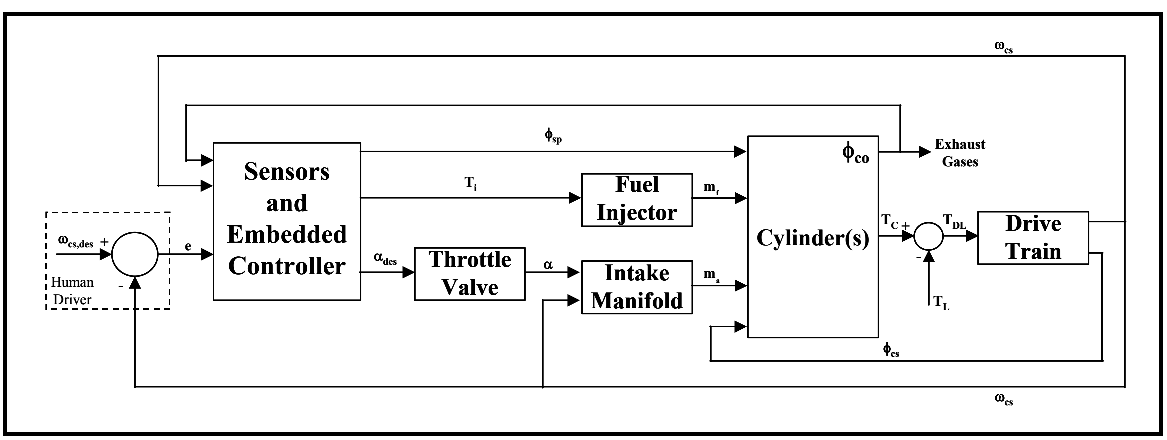
Stroke	$\phi_{co} < 0$ $\left(\lvert\phi_{co}\rvert \le \phi_p \le 720 + \lvert\phi_{co}\rvert\right)$	$\phi_{co} = 0$ $\left(0 \le \phi_p \le 720\right)$	$\phi_{co} > 0$ $\left(-\lvert\phi_{co}\rvert \le \phi_p \le 720 - \lvert\phi_{co}\rvert\right)$
Intake	$720 < \phi_p \le 720 + \lvert\phi_{co}\rvert$ $\lvert\phi_{co}\rvert \le \phi_p \le 180$	$0 \le \phi_p \le 180$	$0 \le \phi_p \le 180$
Compression	$180 < \phi_p \le 360$	$180 < \phi_p \le 360$	$180 < \phi_p \le 360$
Expansion	$360 < \phi_p \le 540$	$360 < \phi_p \le 540$	$360 < \phi_p \le 540$
Exhaust	$540 < \phi_p \le 720$	$540 < \phi_p \le 720$	$540 < \phi_p \le 720 - \lvert\phi_{co}\rvert$ $-\lvert\phi_{co}\rvert \le \phi_p < 0$

Stroke as a function of ϕ_{co} and ϕ_p

Table 1

Engine/Controller Block Diagram

We will model the engine/controller according to the block diagram shown in Figure 3. For simplicity we assume an engine with a single cylinder. The process begins with the human driver shown at the far left in the figure. The driver commands the engine to provide a certain vehicle speed, and he does this with the accelerator pedal. Of course, we don't input a numerical value of speed to the controller when we drive, instead, we sense whether or not we're going at the desired speed and indicate more speed (depress the accelerator pedal) or less speed (let up on it) if we're not. The controller thus receives and *error signal* from the driver, telling it (ultimately) how much torque to deliver to the drive train. This isn't the only objective the controller must satisfy, however. It must also deliver the requested torque in such a way as to minimize vehicle emissions and fuel consumption, and passenger discomfort due to inordinately high acceleration peaks due to the dynamic response of the overall system. For this purpose it also receives information concerning the vehicle's speed and the composition of the exhaust gases. The three command signals that the controller uses to accomplish these objectives are the cylinder spark advance ϕ_{sp}, the fuel injector timing T_i, and the air inlet throttle valve angle α. Each of these will be described and discussed in greater detail below. The torque output



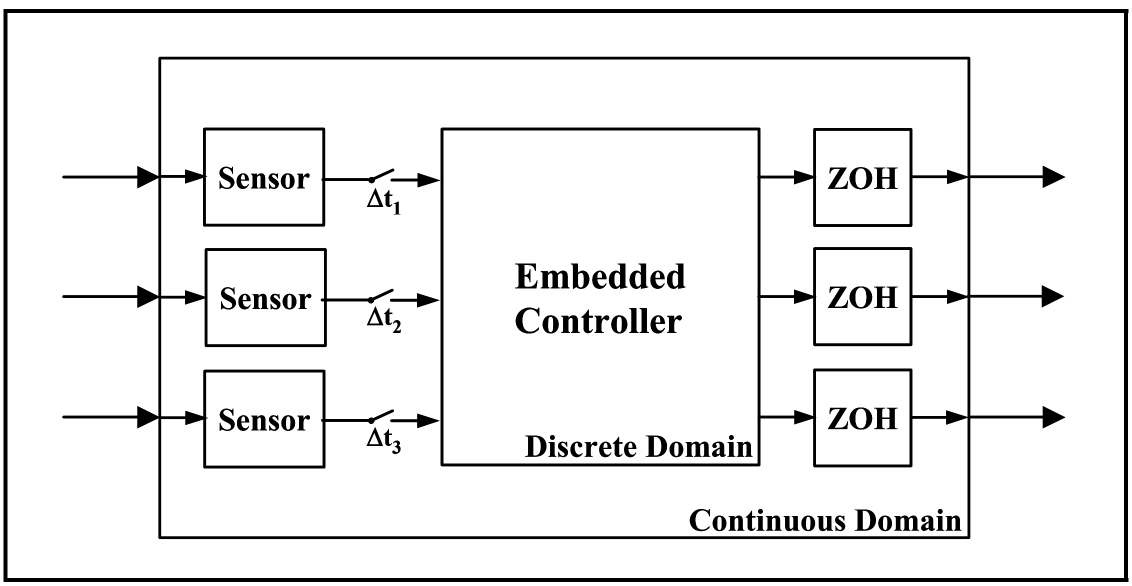
Engine/Controller Block Diagram

Figure 3

of the cylinder T_C is reduced by any load torque T_L that may be present. T_L represents vehicle loads other than that represented by the mass of the empty vehicle, occupied only by the driver, and traveling along a level surface. It can thus represent other passengers, luggage, a trailer, an uphill (or downhill, which constitutes a negative load torque) stretch of road, wind gusts, bumps, and the like. It can also have a random component. The net torque T_{DL} is then input to the drive train, which produces a crankshaft speed (and position) as a result. These are then fed back to the various upstream components of the system. We now consider each of these components in turn.

Sensors and Embedded Controller

The embedded controller receives signals concerning desired speed, actual speed, and exhaust gas composition through sensors, as indicated in Figure 4. The sensors and various engine components are all continuous time devices, but the controller—a digital microprocessor—is typically a discrete time device. For this reason the continuous signals from the sensors must first be converted to discrete signals via a sampler or *event generator*. Each event generator samples with some sampling period Δt_i, as indicated in the figure, which may be different from the other event generators' sampling periods. The discrete control signals that are subsequently output by the controller are then converted back to continuous signals via *Zero Order Holds*, also shown. Note how the discrete domain must be isolated from the continuous domain that it resides in. This is not only an important feature of the model. The computational domain that the model is constructed in must also be capable of providing the proper interfaces to allow the various domains to communicate properly. We'll discuss this in more detail later on.



Sensors and Embedded Controller
Figure 4

Fuel Injector

The fuel injector is a valve, on the upstream of which is liquid fuel under high pressure. Based on the input from the controller, the injector valve opens for a specified time T_i and delivers a mass of fuel to the cylinder during the intake stroke according to

$$m_f = g_i \left(T_i - T_{offset} \right), \tag{3}$$

where

$$\begin{aligned} g_i &= 1.96 \cdot 10^{-3} \, {}^{kg}\!/\!_{sec}, \\ T_{offset} &= 7.5 \cdot 10^{-4} \, \text{sec}. \end{aligned} \tag{4}$$

This experimentally derived relation was taken from [14].

<u>Throttle Valve/Servo Motor</u>

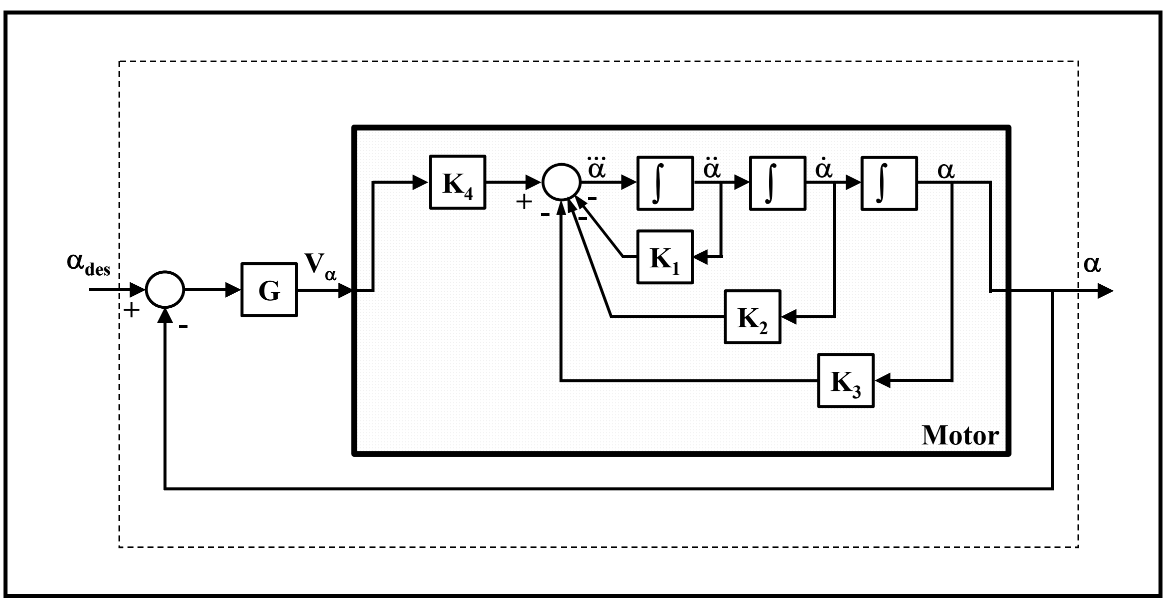
The Throttle Valve/Servo Motor is described by the block diagram in Figure 5. The governing equations for the motor are

$$
\begin{aligned}
&V_b = K_b\,\dot{\alpha}, \\
&V_\alpha = R_\alpha\, i_\alpha + L_\alpha\, \dot{i}_\alpha + V_b, \qquad (5) \\
&T_\alpha = K_t\, i_\alpha = J_\alpha\,\ddot{\alpha} + D_\alpha\,\dot{\alpha} + K_\alpha\,\alpha,
\end{aligned}
$$

where α is the angular position of the motor shaft and thus the butterfly valve connected directly to it ("closed" corresponds to $\alpha = 0°$ and "open" to $\alpha = 90°$). The other terms are defined at the end of this report. Elimination of V_b, V_α, and i_α then gives

$$
\begin{aligned}
\dddot{\alpha} &= -\left[\frac{R_\alpha}{L_\alpha} + \frac{D_\alpha}{J_\alpha}\right]\ddot{\alpha} - \left[\frac{R_\alpha}{L_\alpha} + \frac{K_\alpha}{J_\alpha} + \frac{K_t\,K_b}{L_\alpha\,J_\alpha}\right]\dot{\alpha} + \frac{R_\alpha K_\alpha}{L_\alpha\,J_\alpha}\alpha + \frac{K_t}{L_\alpha\,J_\alpha}V_\alpha, \\
&= -K_1\,\ddot{\alpha} - K_2\,\dot{\alpha} + K_3\,\alpha + K_4\,V_\alpha.
\end{aligned} \qquad (6)
$$

The motor parameters are then chosen to provide the desired performance. The solution of Equ. (6) is represented by the shaded block in Figure 5. Application of an input voltage to a motor of course gives velocity, not position, which we desire. This can be achieved, however, by placing the motor in the simple feedback loop indicated. With this we now have an additional parameter, the feedback gain G, with which to achieve desired transient response and steady state error. (For this problem a satisfactory value for G was determined to be 10 V/deg.)



Throttle Valve/Servo Motor
<u>Figure 5</u>

Intake Manifold

The equations describing the air flow into the cylinder from the intake manifold are presented in [4], [14], and [15]:

$$\dot{p}_{man} = -\frac{V_d}{4\pi(V_d + V_m)} \cdot \eta_v(\omega_{cs}, p_{man}) \cdot \omega_{cs} \cdot p_{man} + \frac{p_{atm}(R_g T_{atm})^{\frac{1}{2}}}{(V_d + V_m)} \cdot \beta\left(\frac{p_{man}}{p_{atm}}\right) \cdot A_{eff}(\alpha)$$

$$m_a = \frac{V_d}{4 R_g T_{atm}} \cdot \eta_v(\omega_{cs}, p_{man}) \cdot p_{man}, \tag{7}$$

where p_{man} is the intake manifold pressure and m_a is the mass of air entrained in the cylinder at the end of the intake stroke. The equations are derived from theoretical principles and modified by the inclusion of a *volumetric efficiency* η_v and an *effective inlet throat area* A_{eff} to account for flow losses. These terms are experimentally derived, where

$$\eta_v = .5497071 + 5.15605\left(\frac{30}{\pi}\omega_{cs}\right) + 9.30417 \cdot 10^{-7} p_{man}, \tag{8}$$

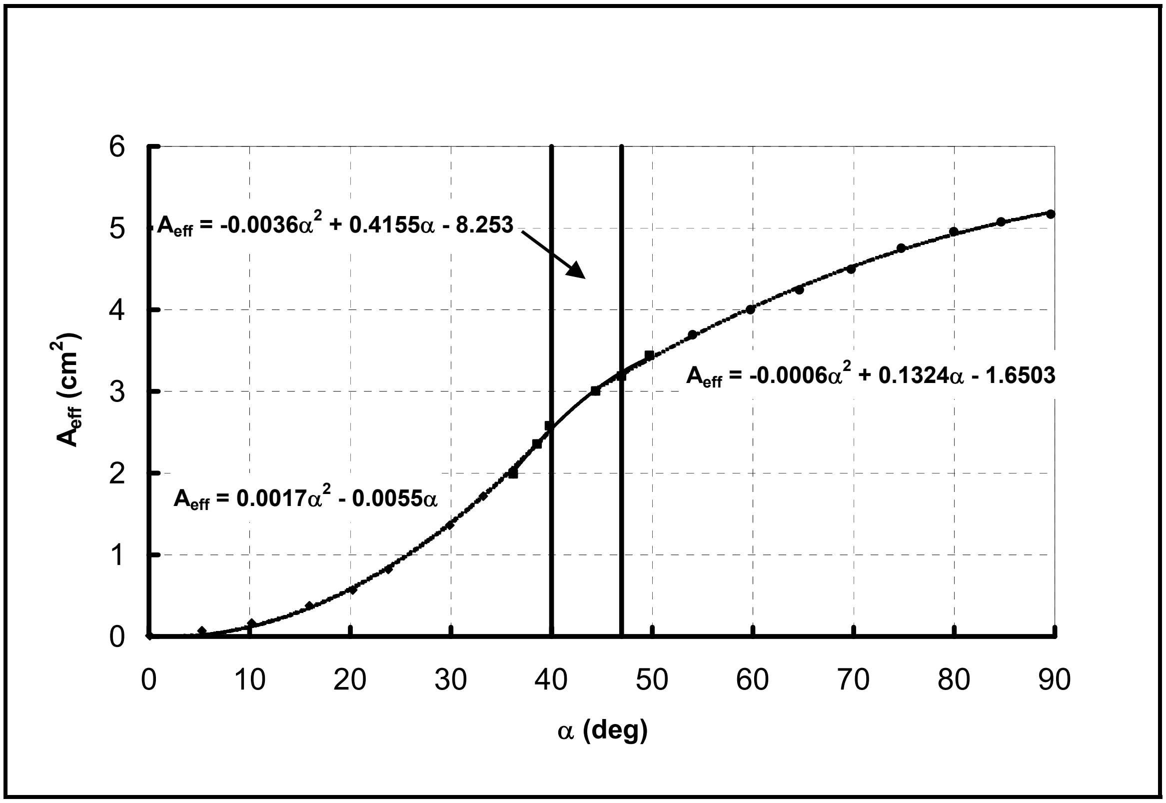
and where A_{eff} is given by Figure 6. β tests for sonic flow, and is given by

$$\beta(y) = \begin{cases} \left[\frac{2\gamma}{\gamma - 1}\left(y^{\frac{2}{\gamma}} - y^{\frac{\gamma+1}{\gamma}}\right)\right]^{\frac{1}{2}} & y \geq r_c \\ \gamma^{\frac{1}{2}}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} & y < r_c \end{cases}, \tag{9}$$

where

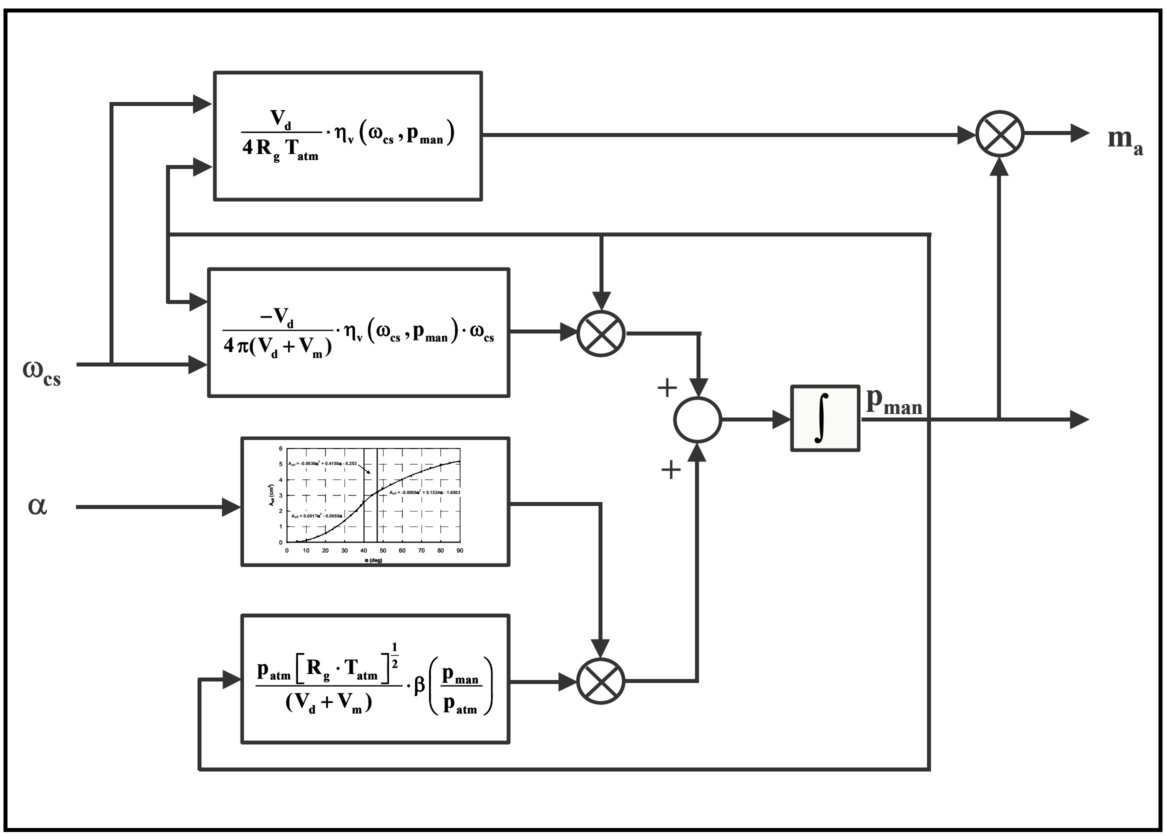
$$y = \frac{p_{man}}{p_{atm}} \qquad r_c = \left[\frac{2}{\gamma+1}\right]^{\frac{\gamma}{\gamma-1}}, \tag{10}$$

and where γ is the ratio of specific heats for air. The other parameters shown are constants for the engine or thermodynamic properties of air and are given at the end of this report. These relations are highly nonlinear, and are evaluated according to the block diagram shown in Figure 7.



Effective Throat Area A_{eff}

Figure 6

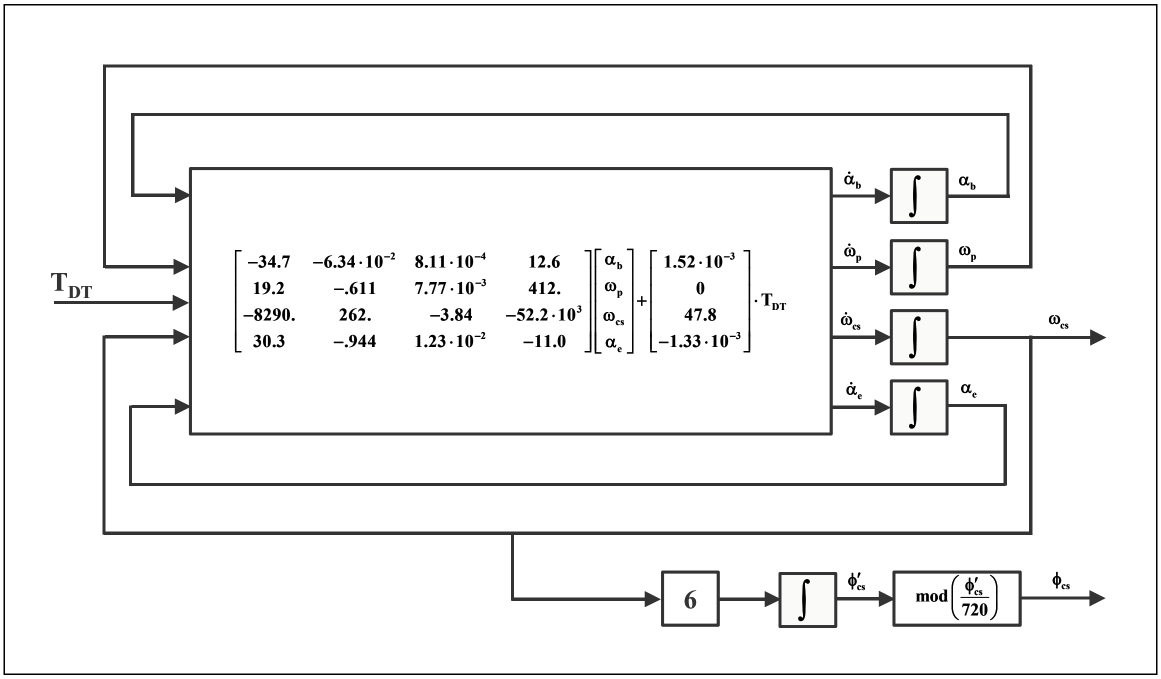


Air Manifold Block Diagram

Figure 7

Drive Train

The drive train is described by the block diagram shown in Figure 8, taken from [9]. It is a 4th order linear system, based on a 7th order nonlinear model of a manual transmission fixed in its highest forward gear. The only state variable required by the other components of the model is the crankshaft speed ω_{cs}, which is then integrated to give crankshaft position ϕ_{cs}, and adjusted so that it varies between 0° and 720°.



Drive Train Block Diagram

Figure 8

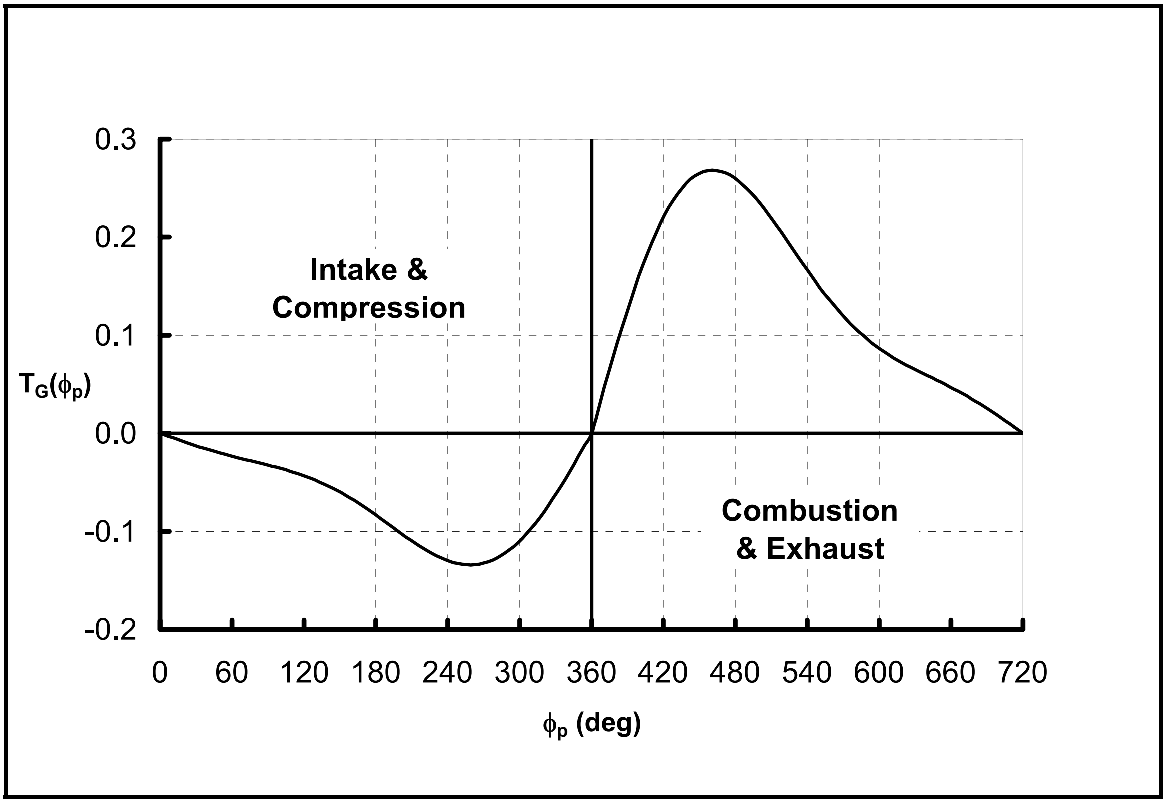
Torque Generation in the Cylinder

Figure 9 shows the "generic" profile of a cylinder's torque output over one cycle, based on experimentally derived data [7]. A Fourier Series has been fitted to this data, the first three harmonics of which are

$$T_G(\phi_p)=\begin{cases} .1\left[\sin\left(\dfrac{\phi_p-2\pi}{2}\right)+\dfrac{1}{2}\sin\left(\phi_p-2\pi\right)+\dfrac{1}{6}\sin\left(\dfrac{3(\phi_p-2\pi)}{2}\right)\right]\le 0 & 0\le\phi_p\le 360 \\ .2\left[\sin\left(\dfrac{\phi_p-2\pi}{2}\right)+\dfrac{1}{2}\sin\left(\phi_p-2\pi\right)+\dfrac{1}{6}\sin\left(\dfrac{3(\phi_p-2\pi)}{2}\right)\right]\ge 0 & 360\le\phi_p\le 720 \end{cases}. \tag{11}$$

Note from this that the total torque for the cycle (the integral under the curve) will always be positive, due to multiplication by .2 in the positive portion of the profile. As mentioned, this is just a generic profile—in order to achieve the actual amount of torque necessary to attain a desired engine speed, Equ. (11) is scaled by a factor T_S. T_S is determined with the aid of another experimentally derived relation [9] and [23] for the overall torque produced during a cycle,

$$T_C = G_T\,(ma+m_f)\,\eta_{ign}(\phi_{sp}), \tag{12}$$



"Generic" Torque Waveform
Figure 9

where

$$G_T = 2.5 \cdot 10^4 \frac{\mathrm{N}}{\mathrm{kg}}, \tag{13}$$

and where η_{ign}, the *ignition efficiency*, is another experimentally derived relation [23] given in Figure 10. Typically, we think of the spark as firing at the start of the expansion stroke ($\phi_p = 360°$). Because of the finite time it takes for the flame front to propagate, however, it is usually desirable to *advance* the spark so that it fires before the completion of the compression stroke. Alternatively, there are other circumstances in which it is desirable to *retard* the spark so that it fires after the beginning of the expansion stroke. The piston position at which the spark fires is conventionally given by

$$\phi_{fire} = \phi_p - \phi_{sp} = 360° - \phi_{sp}, \tag{14}$$

where ϕ_{sp} is the *spark advance*. From this, we see that positive ϕ_{sp} advances the spark into the compression stroke and negative ϕ_{sp} retards it to occur in the expansion stroke. Figure 10 gives the limits on the values ϕ_{sp} can take as well as the effect it has on output torque (through η_{ign}). Combining Equs. (11) and (12) and including the scale factor T_S, gives

$$T_S \int_0^{720} T_G(\phi_p)\, d\phi_p = G_T(ma + m_f)\, \eta_{ign}(\phi_{sp}), \tag{15}$$

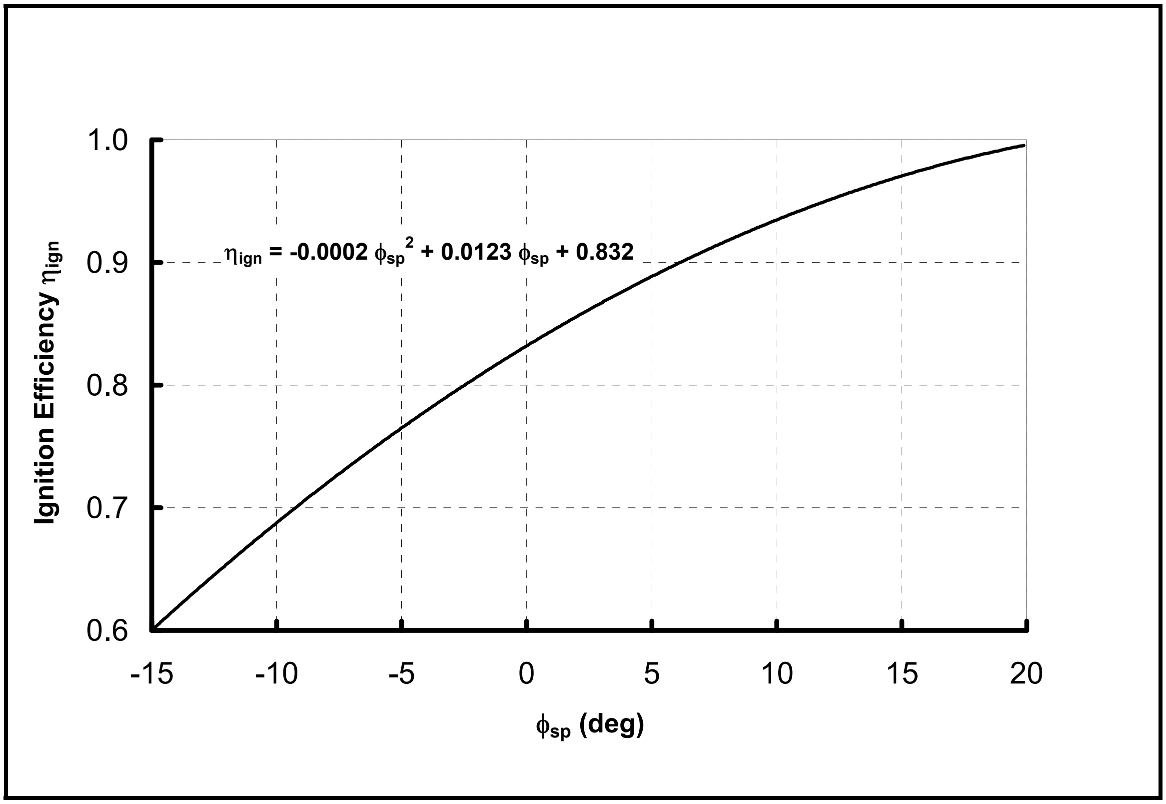
for which

$$T_S = \frac{G_T(ma + m_f)\, \eta_{ign}(\phi_{sp})}{\int_0^{720} T_G(\phi_p)\, d\phi_p}, \tag{16}$$

The integral is easily evaluated, resulting in

$$T_S = \frac{45}{19}\left[G_T(ma + m_f)\, \eta_{ign}(\phi_{sp})\right]. \tag{17}$$

In the model, m_a, m_f, and ϕ_{sp} are sampled once during the intake stroke, and T_S calculated. This is then used to determine the torque output of the cylinder $T_S \bullet T_G(\phi_p)$ for the remainder of that cycle.

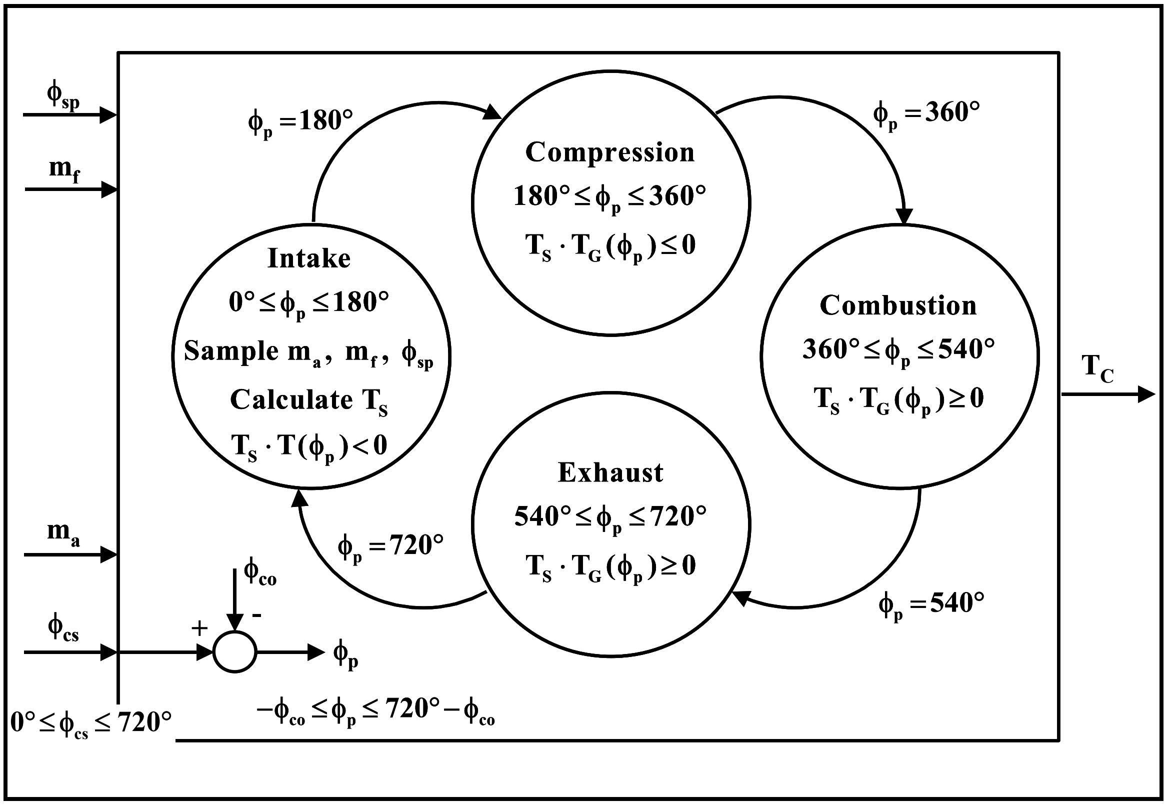


Ignition Efficiency η_{ign}

Figure 10

The Cylinder Module

Previously, we described the cylinder's continuous time, event-based nature. Such systems are modeled as *Finite State Machines (FSM's)*, and a simplified rendition of the cylinder FSM is shown in Figure 11 (where, for clarity, both ϕ_{co} and ϕ_{sp} are assumed to be zero). Transitions between the four strokes are triggered by evolution of ϕ_p, as indicated. During the intake stroke, as just discussed, m_a, m_f, and ϕ_{sp} are sampled and T_S calculated. This module thus gives torque as a function of piston position ϕ_p.



Simplified Cylinder Finite State Machine

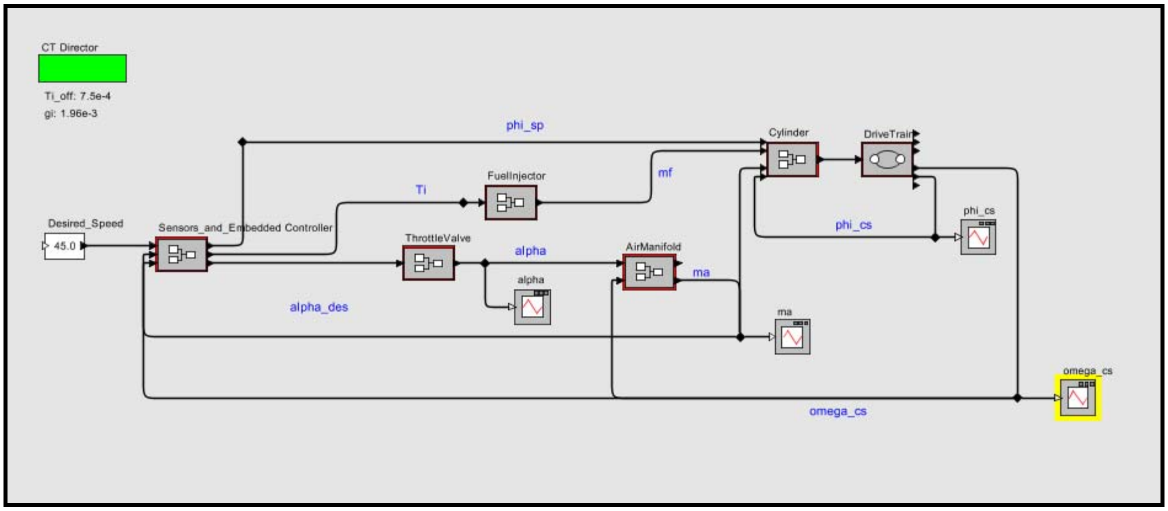
Figure 11

Discussion/Simulation

Implementation of the Model in Ptolemy II

Ptolemy supports *hierarchical models*, which are models containing components that are themselves models. Such components are called *composite actors*. One nice thing about composite actors is that we can switch domains as we go down the hierarchy. A composite actor can exist in, say, the CT domain, but the model inside the actor could be in the DE domain. The embedded controller in our engine model is an example of this (See Figure 4). *Modal Models*, described above, can also be placed inside composite actors. The cylinder module (Figure 11) is an example of a modal model residing in a composite actor. Though Ptolemy II does a lot of the work for us, we nonetheless have to be careful in how we connect composite actors with different domains to one another. Another nice feature of composite actors and modal models is that we can easily change them out with other compatible actors—one of the requirements of our model. For physical models such as ours, the actors operate simultaneously and can have multiple, simultaneous sources of stimuli. They also operate in a timed (real world) environment. This is true not only for the continuous time modules like the air manifold module, but also for the FSM cylinder module and the discrete event controller. The particular choice of domains for the various modules was easy in our case, because they specifically model the real-world environment we wish to represent.

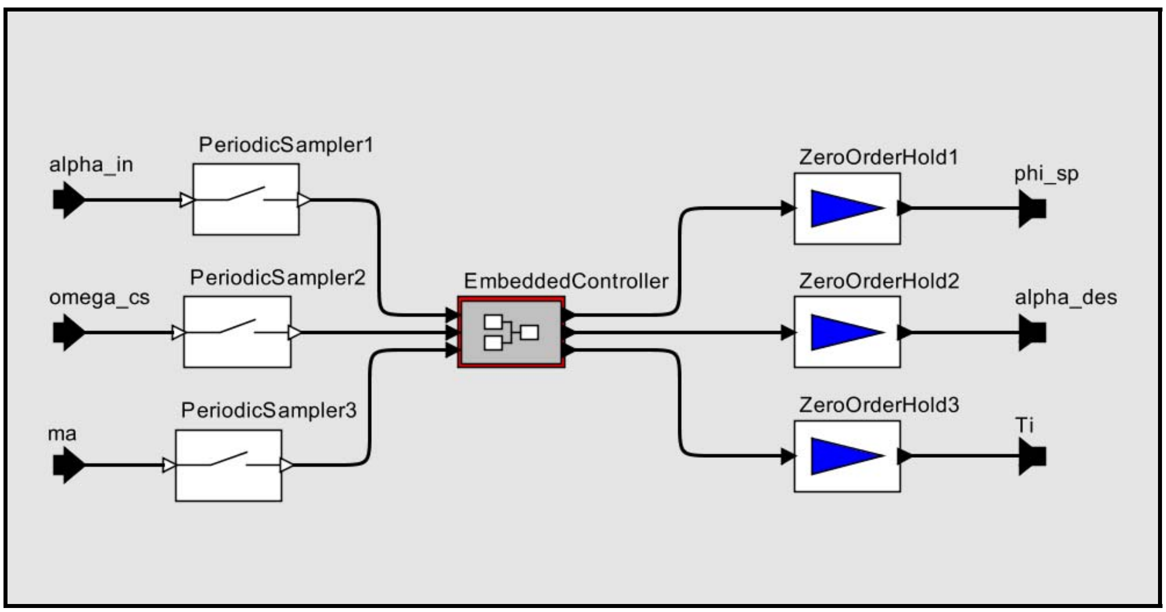
The engine model has been implemented in Ptolemy as an interconnection of composite actors, as shown in Figure 12. This was done to allow for easy interchange and refinement of the individual modules making up the model, as discussed in the previous sections. At this highest level, the CT



Implementation of the Engine Model in Ptolemy

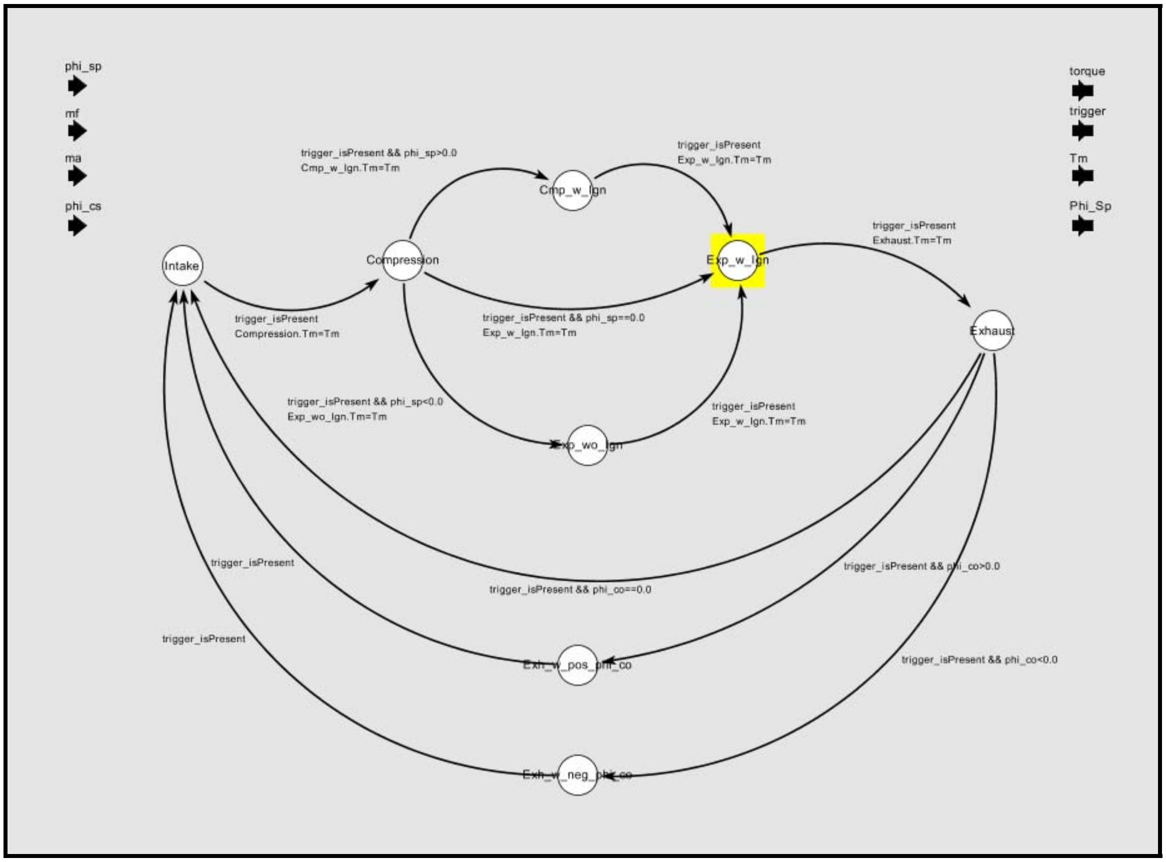
Figure 12

domain is employed. As noted, other domains can be employed inside the individual composite actors, and this has been done in the case of the embedded controller (DE domain) the cylinder (FSM domain), and the drive train (FSM domain). Since all of the composite actors except the embedded controller utilize the CT domain, the interconnections between them were straightforward. More care, of course, had to be taken with the embedded controller, shown in Figure 13, and as discussed previously (See too Figure 4). Here we isolate the DE controller with event generators upstream of it and zero order holds



Embedded Controller

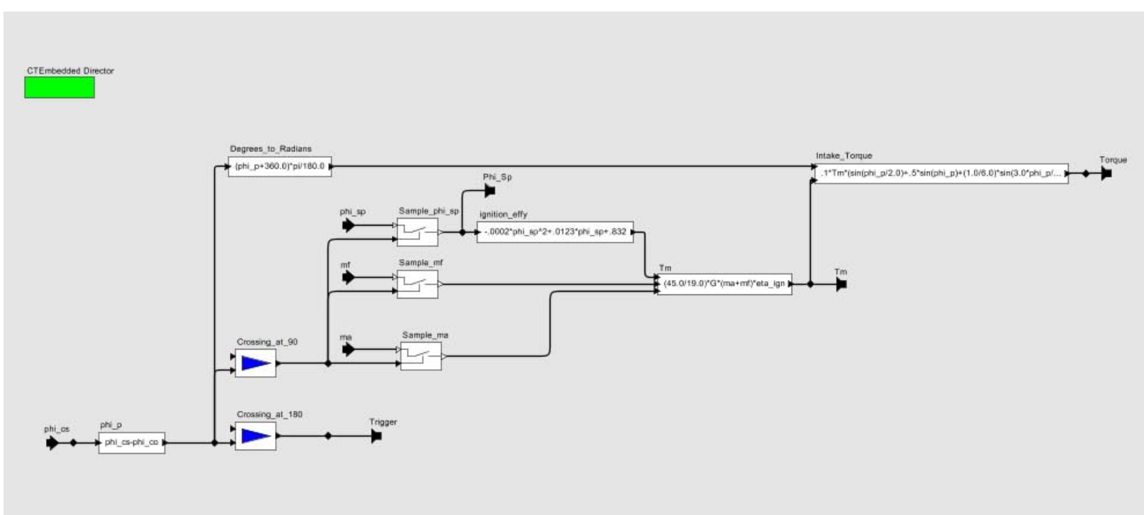
Figure 13



The Cylinder Finite State Machine

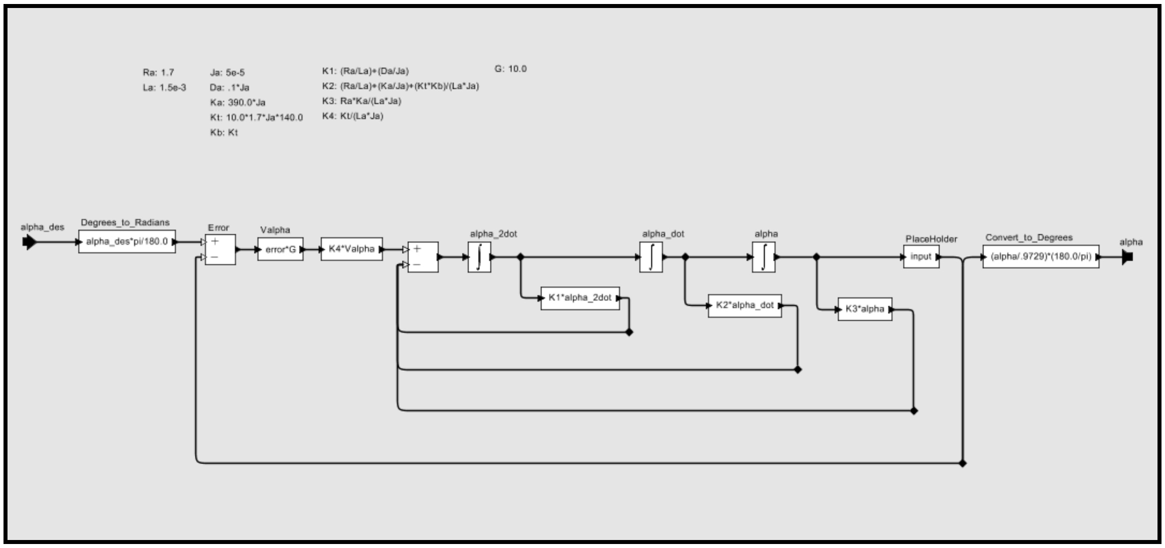
Figure 14

downstream of it. This was the first major subtlety of the implementation. The second had to do with the cylinder module, shown in Figure 14. As described before, the cylinder inputs m_a, m_f, and ϕ_{sp} are sampled once per cycle, during the intake stroke To accomplish this, a level crossing detector set at $\phi_p = 90°$ triggers a sampler, which then samples the inputs. This is shown in Figure 15. The remaining modules were implemented in a very straightforward fashion from the figures already presented. Their implementations are shown in Figures 16-19.



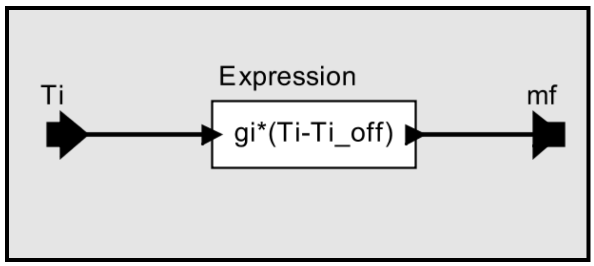
The Intake State

Figure 15



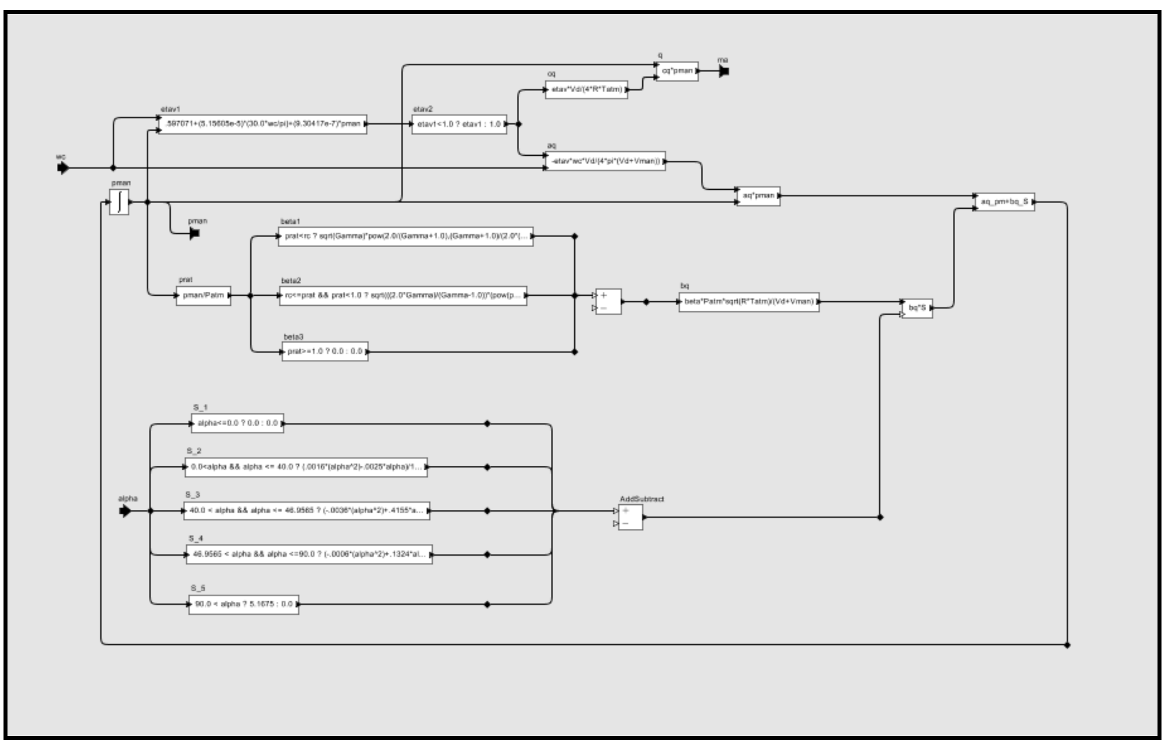
The Throttle Valve

Figure 16



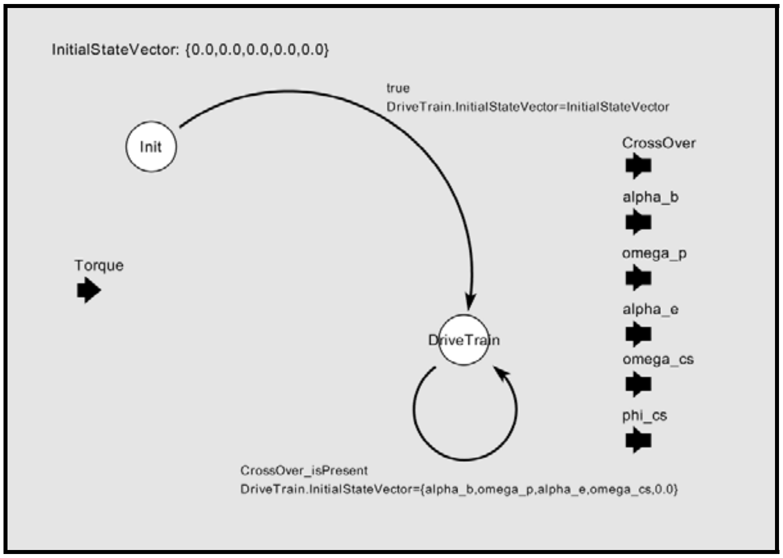
The Fuel Injector

Figure 17

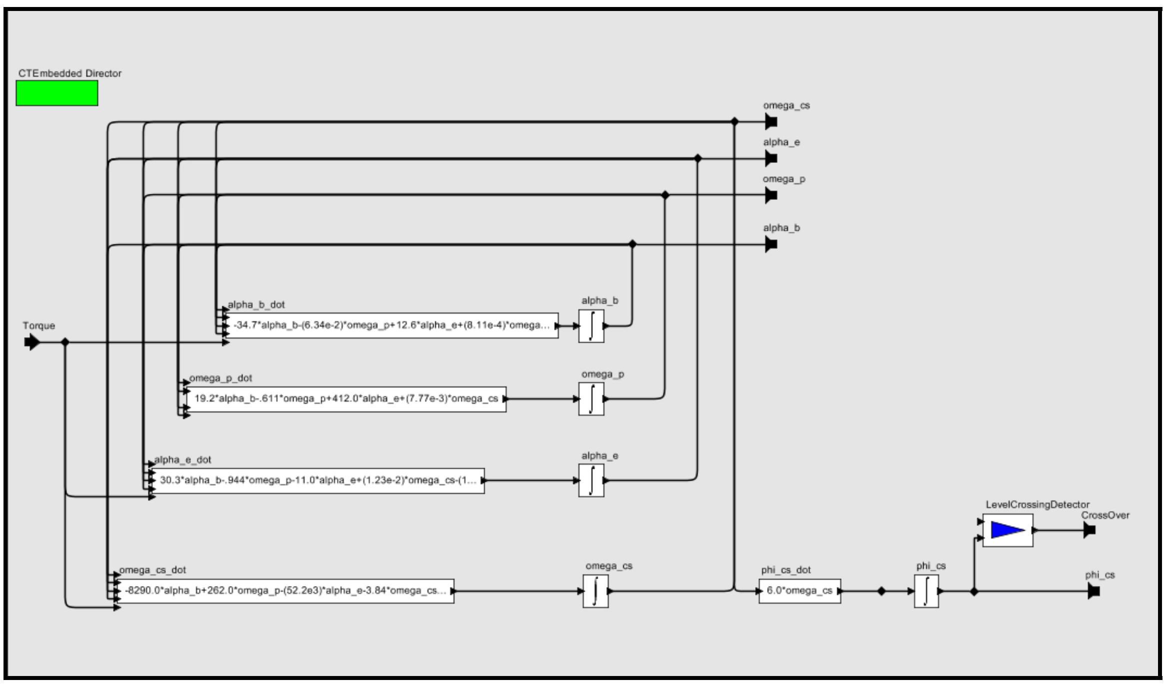


The Intake Manifold

Figure 18



The Drive Train Finite State Machine
Figure 19a

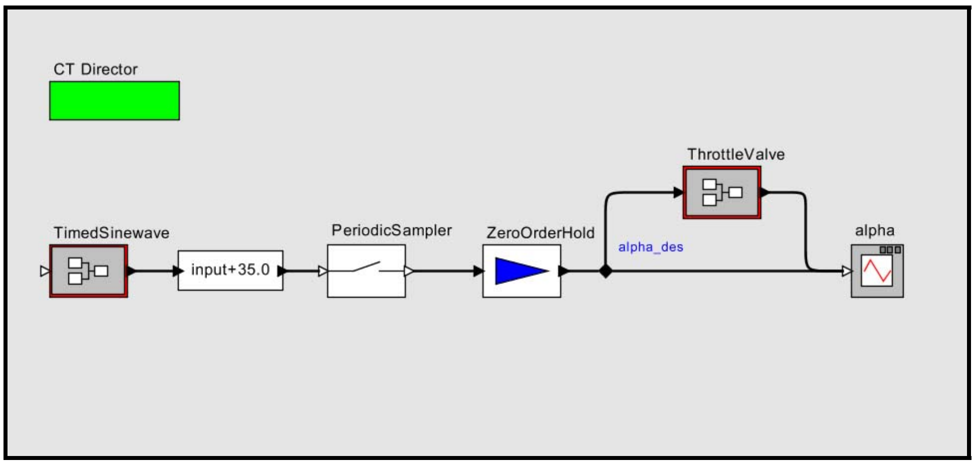


The Drive Train State
Figure 19b

Simulation Results

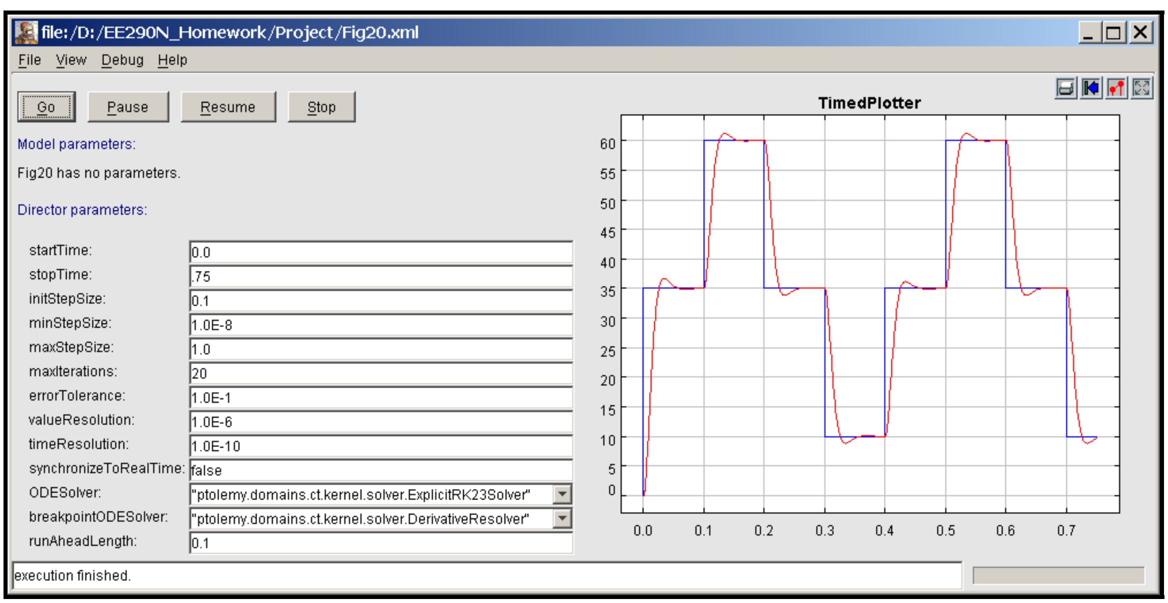
Ptolemy is still under development, and modifications made to it by the development team unfortunately introduced bugs that prevented the controller and cylinder from being implemented, and also prevented the overall engine from being constructed and simulated in time to be reported here. In place of this, then, we will demonstrate the performance of the individual modules making the model up, as well as several combinations of components. We begin with the Throttle Valve, which has been placed in the circuit shown in Figure 20a. The upstream components are intended to simulate a discrete controller, inputting for α_{des} a staircase sine wave. The Throttle Valve's performance is shown in Figure 20b. Note the fast and accurate response to an input that it would never experience in actual practice. The Intake Manifold, shown in Figure 21a, is driven by a simulated Throttle Valve that is opening in steps of 20°, and a crankshaft that is undergoing a sinusoidal variation in angular velocity. Again, these are extreme inputs. Its performance is shown in Figure 21b. The Drive Train is shown in Figure 22a, and its response to a sinusoidally varying input torque, which changes sign, is shown in Figure 22b. In Figure 23a we combine the cylinder and the drive train. Here we assume that both ϕ_{co} and ϕ_{sp} are zero, and that m_a and m_f are constant at the stoichiometric ratio (not shown). The drive train is driven by a constant torque. Figure 24b shows the torque output of the cylinder, which compares favorably to the waveform of Figure 9.

We end by combining the Throttle Valve, Intake Manifold, Fuel Injector, and Drive Train together, as shown in Figure 24a. Here we supply the same inputs to the Throttle Valve and the Drive Train as in the previous cases, but now the Throttle Valve's output forms the input to Intake Manifold. The Simple Controller senses m_a from the intake manifold and computes the input to the Fuel Injector, T_i, to give the stoichiometric amount of fuel. The Controller is shown in Figure 24b. To simulate its discrete nature, we convert m_a to a train of pulses prior to computing T_i. The results are plotted in Figure 24c. Note in particular how the Air/Fuel ratio fluctuates due to the discrete sampling. (Prior simulations which excluded the sampler and treated the input as a continuous signal gave the exact stoichiometric ratio, as one would expect.)



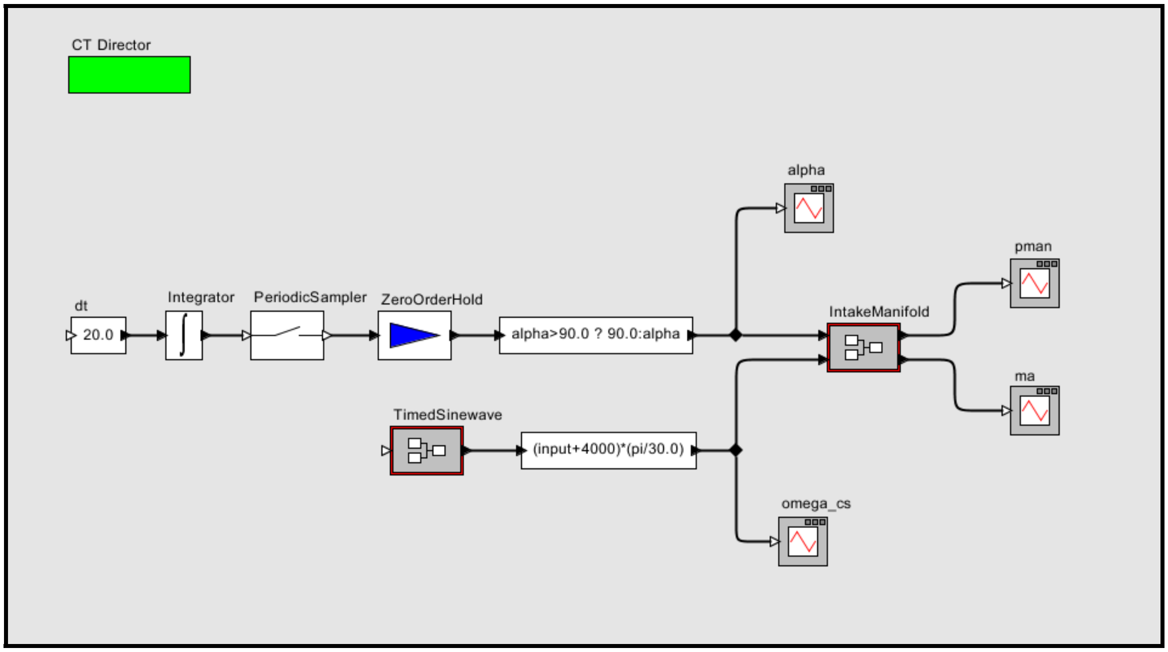
Throttle Valve Simulation

Figure 20a

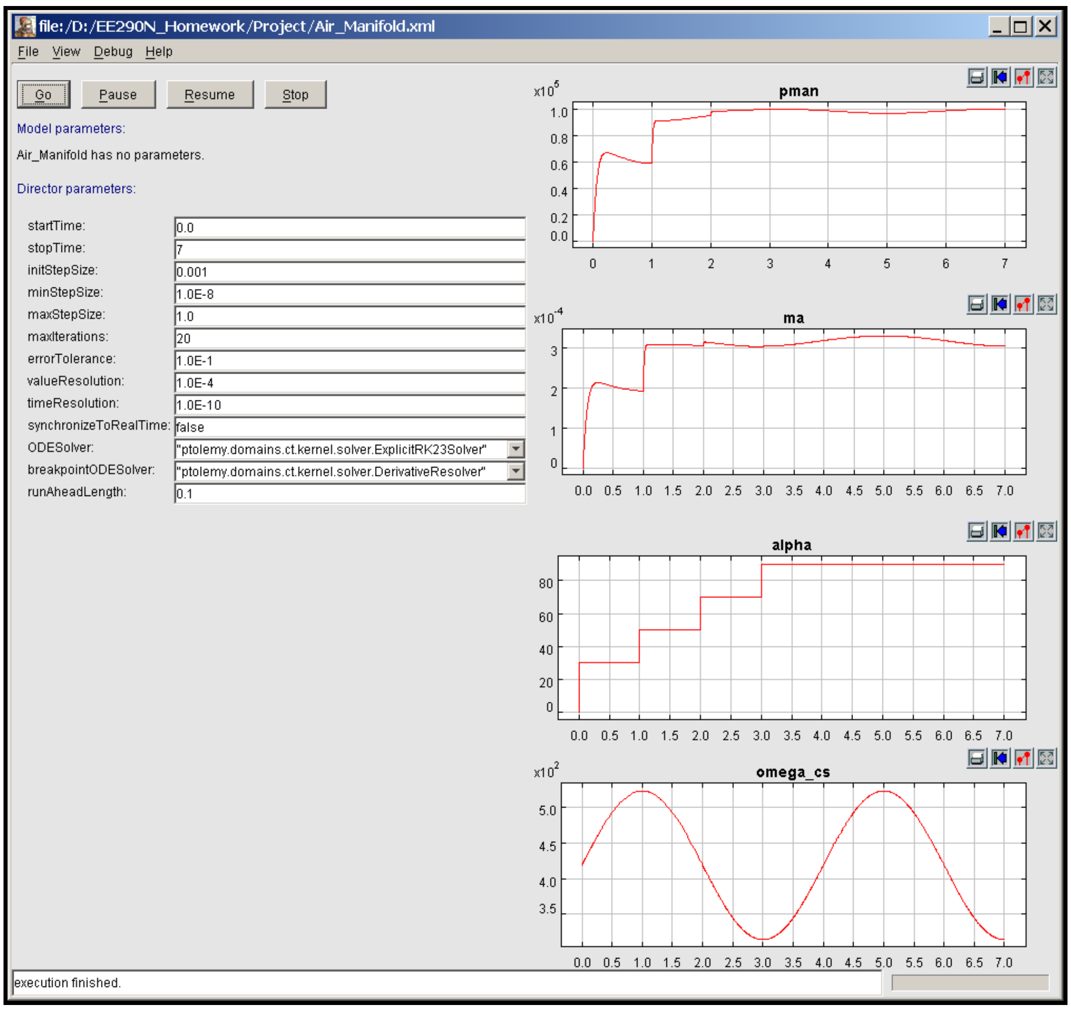


Throttle Valve Simulation

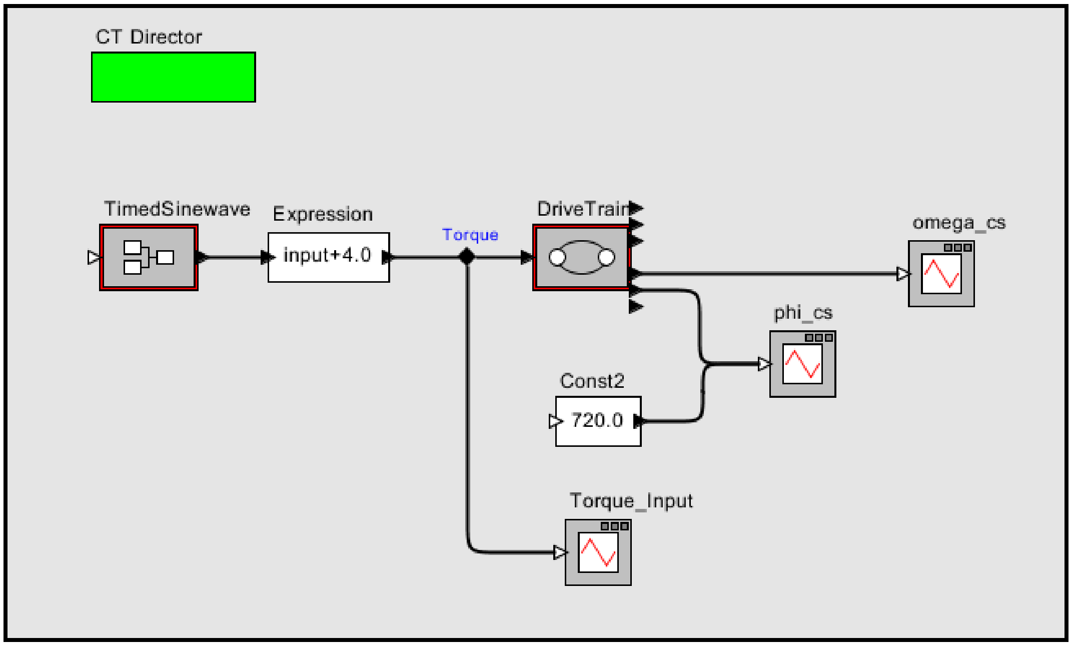
Figure 20b



Intake Manifold Simulation
Figure 21a

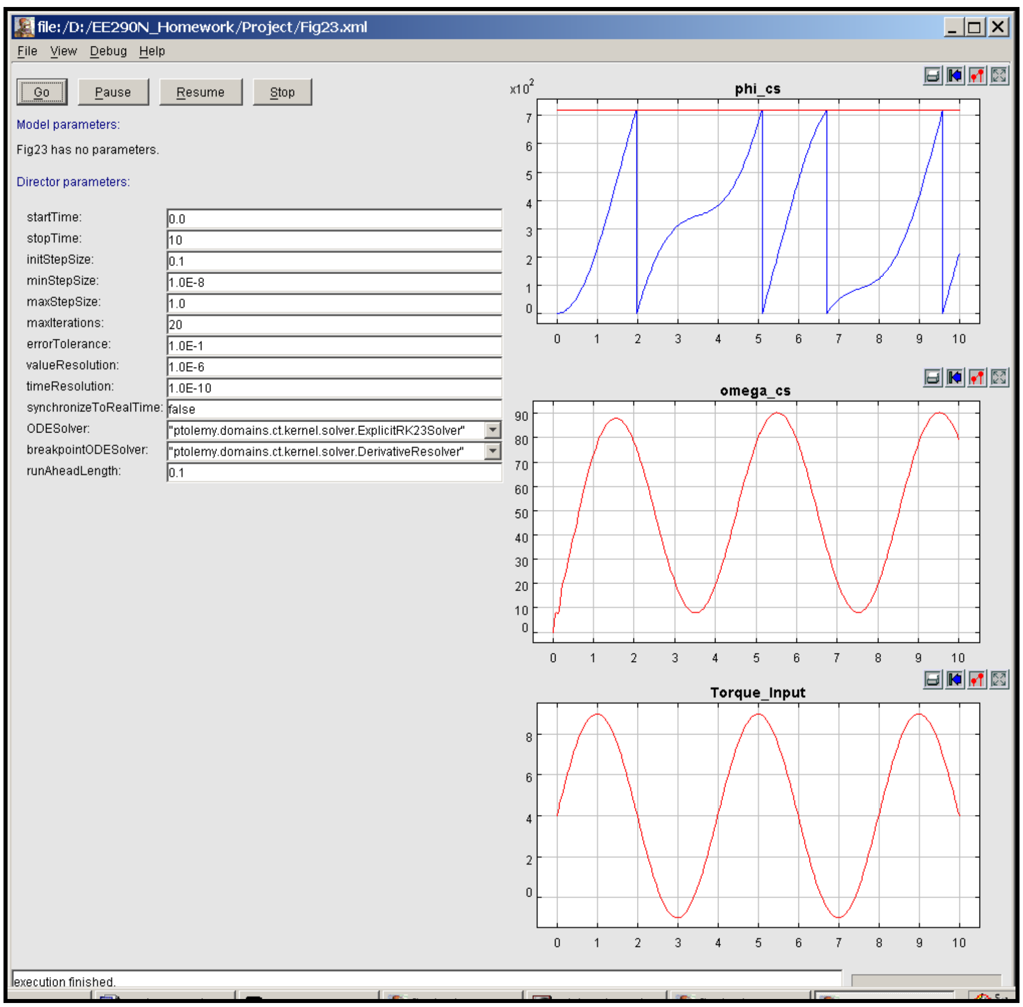


Intake Manifold Simulation
Figure 21b



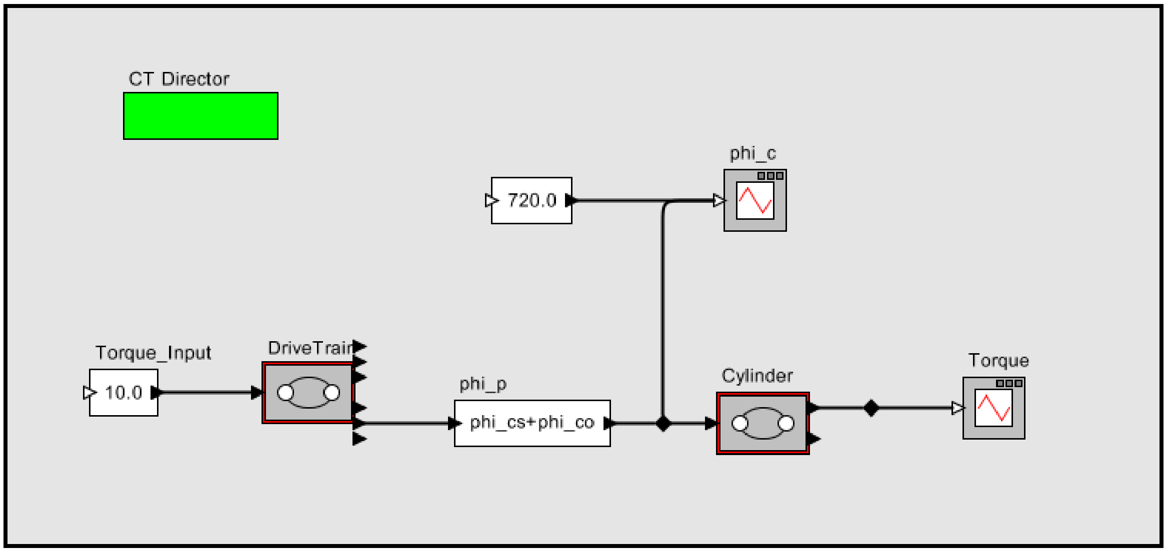
Drive Train Simulation

Figure 22a

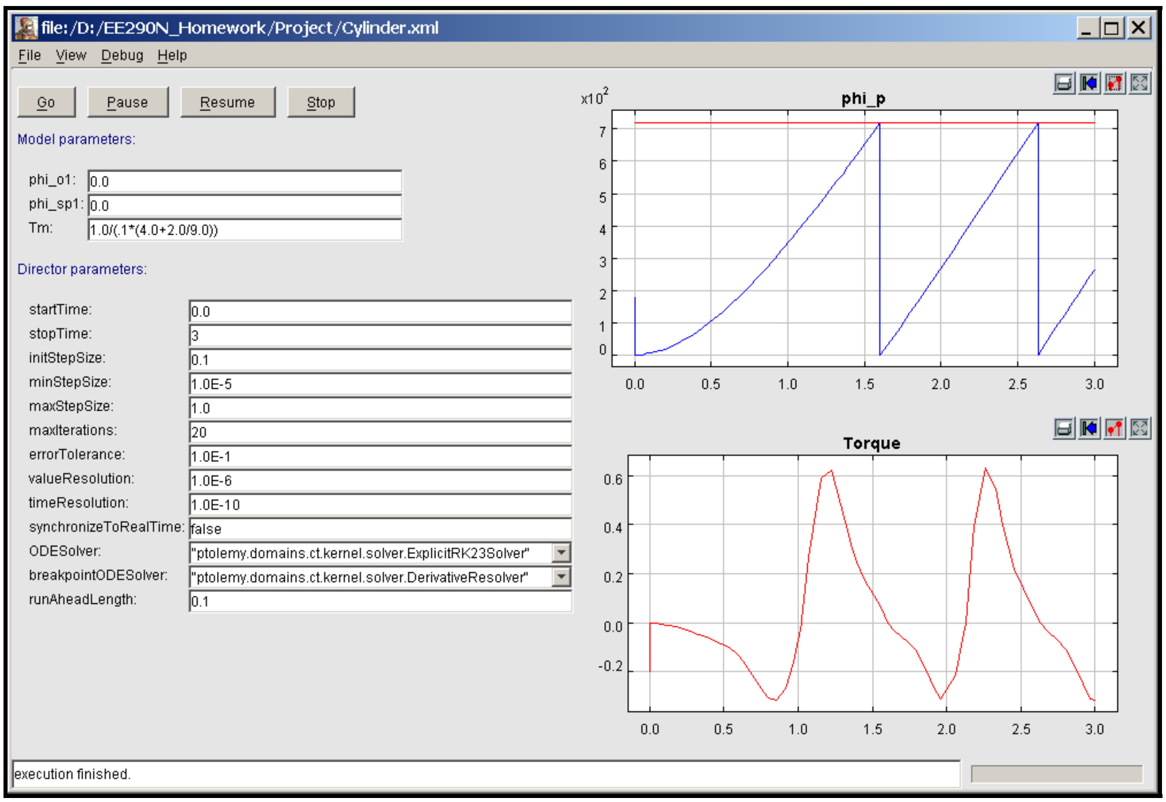


Drive Train Simulation

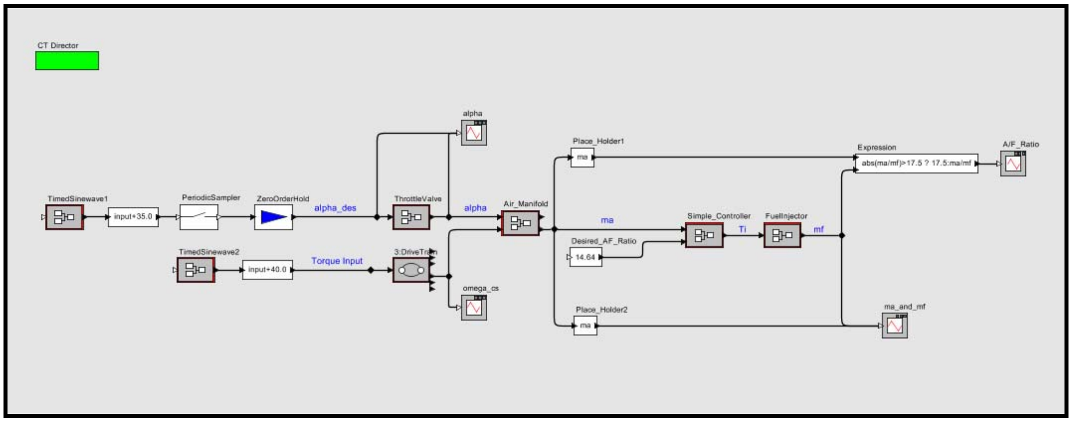
Figure 22b



Cylinder/Drive Train Simulation
Figure 23a

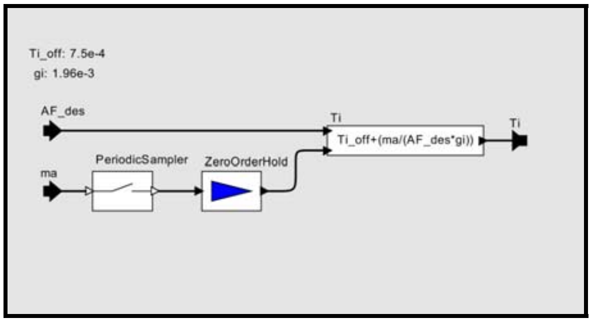


Cylinder/Drive Train Simulation
Figure 23b



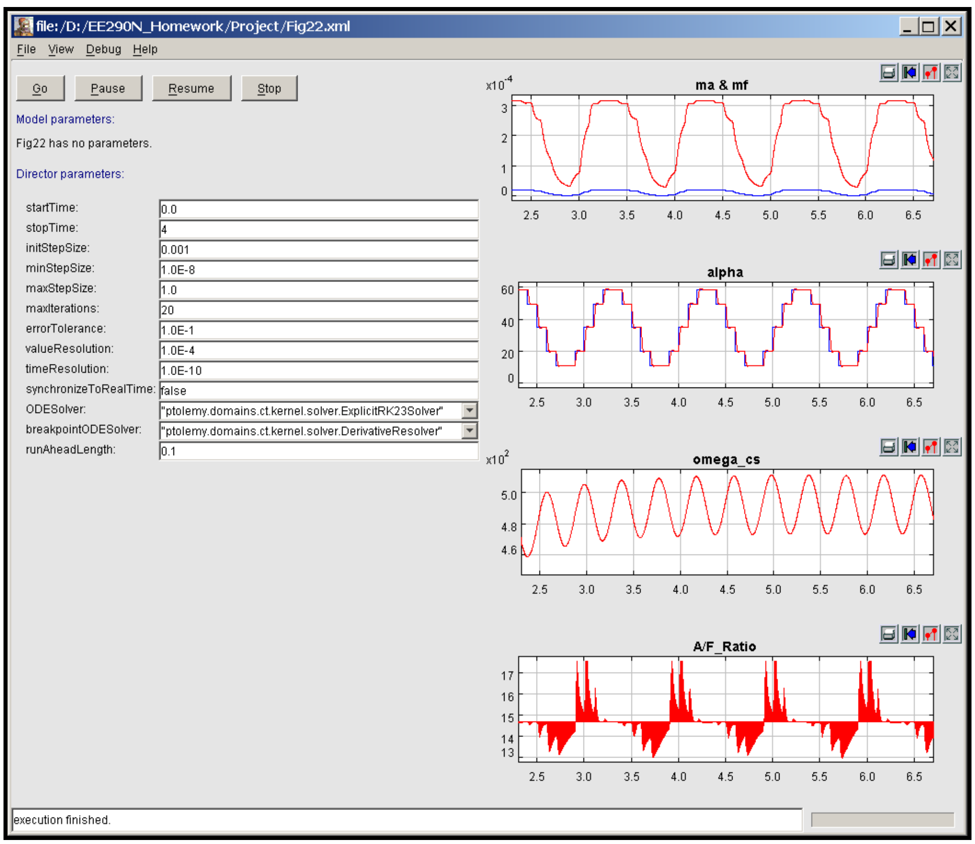
Throttle Valve, Intake Manifold, Fuel Injector, and Drive Train

Figure 24a



Simple Controller

Figure 24b



Throttle Valve, Intake Manifold, Fuel Injector, and Drive Train

Figure 24c

Conclusions

A simplified model of an internal combustion engine was developed in the Ptolemy environment, demonstrating the applicability of the *component-based design* approach to this problem. Problems encountered with Ptolemy, which is still under development, precluded the construction and verification of the complete model, but the performance of individual components and several combinations of components was simulated and demonstrated. Key features of the model are its modular construction, permitting individual aspects of the engine to be abstracted or refined, depending on the problem at hand; and the fact that different modules are modeled in different domains, permitting study of the interaction of a continuous time physical system with a discrete event controller. The final simulation (Figure 22c) showed how the system's behavior can change when such distinctions are incorporated.

The model developed here is generally based on the previously cited references, but it departs from those efforts most notably with respect to the cylinder module. The references model the cylinder in a discrete, event-based domain. Torque output is zero during the intake, compression, and exhaust strokes, and some constant net value during the expansion stroke. In contrast, in this project the cylinder is modeled in the continuous domain (though as a finite state machine), and provision is made to provide a realistic torque output during every stroke of the piston. Inclusion of additional cylinders, offset with respect to one another as they would be in a real engine, should therefore give a very realistic, *instantaneous* torque output. The incorporation of the servomotor driven inlet valve is a second innovation of this model.

It remains to incorporate a realistic model of the controller, using as a guide the previously cited references. Sensor dynamics and noise contamination usually associated with their operation should also be undertaken, and would be very easy to do in this environment. In order to stay focused on demonstrating the method, very simple models of torque generation in the cylinder were used. More sophisticated models are readily available, and this should be the next task undertaken after the controller and sensors. Next in line would be the fuel injector, which could be modeled much more accurately, then the drive train (to incorporate important non-linearities associated with its operation), and finally, another pass at the intake manifold could be taken.

Provision was made for a load torque, but nothing was done with it in this current effort. Many different types of realistic loads could be modeled with it, and these could be developed once the above-mentioned refinements were completed.

Terms and Definitions

Fuel Injector

m_f: mass of fuel injected into the cylinder during an intake stroke (kg)

g_i: fuel injector gain constant $\left(1.96 \cdot 10^{-3}\ {}^{kg}\!/\!_{sec}\right)$

T_i: fuel injection duration (sec)

T_{offset}: fuel injection duration offset $\left(7.5 \cdot 10^{-4}\ \text{sec}\right)$

Throttle Valve/Servo Motor

V_b: servo motor back emf voltage (volts)

K_b: servo motor back emf constant $\left(\dfrac{\text{volt} - \text{sec}}{\text{amp}}\right)$

α: servo motor and throttle valve shaft angle (rad)

V_α: servo motor armature voltage (volts)

R_α: servo motor armature resistance (1.7 ohms)

i_α: servo motor armature current (amp)

L_α: servo motor armature inductance $(1.5 \cdot 10^{-3}\ \text{H})$

T_α: servo motor armature torque (N)

K_t: servo motor torque constant $\left(.1190 \dfrac{\text{N}}{\text{amp}}\right)$

K_b: servo motor back-emf constant $\left(.1190 \dfrac{\text{volt} - \text{sec}}{\text{rad}}\right)$

J_α: servo motor and throttle valve rotational inertia referred to the motor shaft $\left(5 \cdot 10^{-5} \dfrac{\text{N} - \text{m} - \text{sec}^2}{\text{rad}}\right)$

D_α: servo motor and throttle valve damping resistance referred to the motor shaft $\left(5 \cdot 10^{-6} \dfrac{\text{N} - \text{m} - \text{sec}}{\text{rad}}\right)$

K_α: servo motor and throttle valve spring coefficient referred to the motor shaft $\left(.0195 \dfrac{\text{N} - \text{m}}{\text{rad}}\right)$

$K_{1,2,3,4}$: servo motor/throttle valve constants [see Equ. (6)]

G: servo motor gain $\left(10 \dfrac{\text{volt}}{\text{deg}}\right)$

Intake Manifold

p_{man}: manifold pressure $\left({}^{N}\!/\!_{m^2}\right)$

m_a: mass of air injected into the cylinder during an intake stroke (kg)

V_d: cylinder displacement volume $\left(1.242 \cdot 10^{-3}\ \text{m}^3\right)$

V_m: manifold volume $\left(1.500 \cdot 10^{-3}\ \text{m}^3\right)$

η_v volumetric efficiency [see Equ. (8)]

p_{atm}: manifold inlet pressure $\left(\text{assumed STP: } 101{,}325\ {}^{N}\!/\!_{m^2}\right)$

T_{atm}: manifold inlet temperature $\left(\text{assumed STP: } 300\ \text{K}\right)$

R_g: universal gas constant for air $\left(287.0\ \dfrac{\text{J}}{\text{kg-K}}\right)$

A_{eff}: effective inlet throat area (m^2, see Figure 5)

C_p: constant pressure specific heat for air $\left(1.005 \ \frac{\text{J}}{\text{kg-K}}\right)$

C_v: constant volume specific heat for air $\left(.718 \ \frac{\text{J}}{\text{kg-K}}\right)$

γ: ratio of specific heats for air (1.400)

Drive Train

T_{DT}: drive train input torque (N) (see Figure 3)

α_b engine block angle (rad)

ω_p: wheel speed (rad/sec)

ω_{cs}: crankshaft speed (rad sec)

α_e: axle torsion angle (rad)

ϕ'_{cs}: crankshaft angle (rad) $\left(0 \le \phi'_{cs} \le \infty\right)$

ϕ_{cs}: $\left(\phi'_{cs} \text{ rad} \cdot \frac{180°}{\pi \text{ rad}}\right)$ modulo 720° $\left(0° \le \phi_{cs} \le 720°\right)$

Torque Generation

$T_G(\phi_p)$: generic cylinder torque output (N) [see Figure 8 and Equ. (11)]

T_C: total cylinder torque output per cycle $\left(\text{N}/\text{cycle}\right)$ [see Equs. (12) and (15)]

T_S: cylinder torque scaling factor [see Equs. (16) and (17)]

T_L: load torques in addition to that represented by empty vehicle and driver (N) (see Figure 3)

G_T: Torque Generation Gain $\left(10 \ \text{N}/\text{kg}\right)$

Cylinder

ϕ_p: piston position (rad) [see Equ. (1)]

ϕ_{co}: piston offset angle (rad) (see Figure 2)

ϕ_{sp}: spark advance (rad)